Enhancing AAC Software for Dysarthric Speakers in e-Health Settings: An Evaluation Using TORGO

Macarious Hui*, Jinda Zhang*, Aanchan Mohan*†
*Northeastern University, Vancouver, BC, Canada
{hui.mac, zhang.jinda1,aa.mohan}@northeastern.edu
†University of Victoria, Victoria, BC, Canada

***Abstract*—Individuals with cerebral palsy (CP) and amyotrophic lateral sclerosis (ALS) frequently face challenges with articulation, leading to dysarthria and resulting in atypical speech patterns. In healthcare settings, coomunication breakdowns reduce the quality of care. While building an augmentative and alternative communication (AAC) tool to enable fluid communication we found that state-of-the-art (SOTA) automatic speech recognition (ASR) technology like Whisper and Wav2vec2.0 marginalizes atypical speakers largely due to the lack of training data. Our work looks to leverage SOTA ASR followed by domain specific error-correction. English dysarthric ASR performance is often evaluated on the TORGO dataset. Prompt-overlap is a well-known issue with this dataset where phrases overlap between training and test speakers. Our work proposes an algorithm to break this prompt-overlap. After reducing prompt-overlap, results with SOTA ASR models produce extremely high word error rates for speakers with mild and severe dysarthria. Furthermore, to improve ASR, our work looks at the impact of n-gram language models and large-language model (LLM) based multi-modal generative error-correction algorithms like Whispering-LLaMA for a second pass ASR. Our work highlights how much more needs to be done to improve ASR for atypical speakers to enable equitable healthcare access both in-person and in e-health settings.**

I. Introduction

Healthcare professionals rely on augmentative and alternative communication (AAC) software to support telehealth and in-person appointments for patients with cerebral palsy (CP) and amyotrophic lateral sclerosis (ALS) [1]. Damage to the nervous system can result in paralysis or weakness of the muscles responsible for speech, leading to dysarthria and atypical speech patterns in individuals with ALS or cerebral palsy. Atypical speakers who are verbal, often prefer to use their own voice to communicate their needs. Modern AAC applications like VoiceItt[1] or our own AAC application SpeakEase [2] allow for audio input from the speaker with the intention to provide a faithful transcription. Mulfari et al. [3] propose a low-power, on-device, deep-learning based isolated word ASR system to work in an "always-on" mode for dysarthric speakers with reduced mobility. Such a system has promise to enable communication in healthcare and home settings.

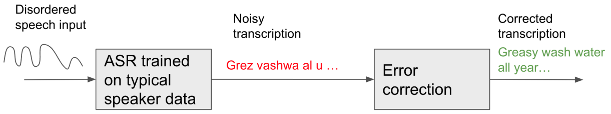

Fig. 1: Dysarthric automatic speech recognition followed by error correction

There is little or no data available on the open domain for atypical speakers. On the other hand web-scale speech datasets like Mozilla Common Voice [4] and GigaSpeech [5] allow for state-of-the-art speech recognition for typical speakers. Dysarthric speech recognition is a low-resource out-of-domain problem [6]. To leverage well-developed typical speaker ASR systems, our work looks for a first pass transcription from such a system followed by error-correction (EC) as show in Figure 1. The figure shows disordered input speech with an imperfect noisy transcription after ASR followed by error-correction. ASR systems are trained with audio and correct transcription pairs. EC systems are trained with inputs consisting of multiple hypotheses of transcribed text (referred to as n-best lists), possibly with errors, with outputs mapping to the correct target text.

1	Ref: he slowly takes a short walk in the open air each day ASR: he shlly takes a wall in the week a eh day EC: he slowly takes a short walk in the open air each day
2	Ref: usually minus several buttons ASR: usually min sell fold buttons EC: usually sell fold buttons
3	Ref: you wished to know all about my grandfather ASR: u' wal awarke youar gread fap EC: you wished to know all about my grandfather

Fig. 2: Inference samples for error-correction (EC) for speaker M05. Ref shows the reference transcription, ASR shows the transcription output which serves as input to the EC model. Notice how the EC system has memorized transcripts due to prompt overlap in TORGO.

To evaluate English ASR for dysarthric speakers, a well-known dataset called the TORGO dataset [7] is widely used. The TORGO dataset for dysarthric speech has data from speakers with either ALS or CP. Other dysarthric ASR databases such as the Nemours corpus[2] [8], UASpeech [9]

[1] www.voiceitt.com

[2] The authors were unable to obtain a recent copy of this database due to a lack of information on the internet

and the HomeService corpus [10] are either hard to obtain or largely consist of isolated word utterances. The TORGO dataset is one of the few containing both isloated word and a few sentence level utterances.

Figure 2 shows inference examples from our initial experiments of error-correction(EC) following ASR. The EC model memorizes the target transcription without doing any error-correction. This issue stems from the dataset design, which features a significant amount of prompt overlap among the speakers. The research community acknowledges that the TORGO dataset has a very high degree of prompt overlap between speakers [11], [12]. This data leakage prevents the dataset from being used to evaluate ASR and EC algorithms for real-world applications like telehealth and e-health.

Our work in this paper makes the following contributions:

- Develop an algorithm based on mixed-integer linear programming to partition the TORGO dataset with no prompt overlap with the constraint to minimize data loss. This dataset is called no-prompt overlap TORGO or NP-TORGO.
- Understand the impact of removing prompt overlap on dysarthric ASR performance using SOTA baseline ASR models.
- Understand the impact of out-of-domain language modelling using text data from the training utterances from NP-TORGO, and Librispeech [13] .
- Understand the impact of error-correction (EC) without ASR system fine-tuning, with a state-of-the-art cross-modal error-correction system such as Whispering-LLAMA [14].

This paper is organized as follows. Section II puts our current work in the context of prior work. Section III introduces the TORGO dataset, and Section IV introduces our approach to remove prompt overlap. Section V presents our experimental setup, and experimental results are presented in Section VI. Section VII provides a discussion of our work, and Section VIII concludes the paper.

II. Relation to prior work

Dysarthric ASR using the TORGO dataset has been well studied in the literature. An ASR system in general consists of an decoder that uses an acoustic model and a language model to return a hypothesized word string. For acoustic modelling, Espana-Bonet et al. [15] looked at the impact of deep neural network based acoustic models on the TORGO dataset. Joy et al. [16] look at different configurations of acoustic models to suggest improvements. Furthermore, Hermann et al. [11] study the impact of lattice-free MMI and how it compensates for speakers with slow speaking rates. Yue et al. [12] state that results for language models trained on training transcripts overestimate ASR results on TORGO due to prompt overlap and investigate the use of out-of-domain language models. In our early experiments on EC for dysarthric ASR, our models started to memorize prompts. Prompt overlap in TORGO is a serious issue we tackle first, before understanding the impact of EC.

To the best of our knowledge little or no work exists on the impact of error-correction following first-pass ASR for atypical speakers. For typical speech, sequence-to-sequence models for error-correction [17] have shown to improve ASR performance. Furthermore, Leng at al. [18] incorporate efficient approaches to consider multiple different hypotheses for EC using transformer based encoder-decoder models. More recently cross-modal EC systems based on large-language models (LLMs) such as Whispering-LLaMA [14] have shown promise. Li et al. [6] study the impact of cross-modal EC using discrete-speech units and their impact on LR-OOD tasks. In the modern deep learning literature Park et al. [19] use a sequence-to-sequence EC module on Korean atypical speech data after using a commodity ASR transcription system. Similar to our work theirs is an AAC application to help children with language disabilities. In contrast to their work, our work looks at the impact of modern multi-modal generative error-correction techniques for English dysarthric ASR.

III. The TORGO dataset

This section briefly introduces the TORGO dataset, and highlights the issue of prompt overlap. Furthermore this section talks about the leave-one-speaker out protocol for TORGO that requires speaker specific train and test sets.

The TORGO data set consists of eight speakers each with varying levels of dysarthria. The database also consists of 7 control speakers. There is about 15 hours of audio from all of the speakers, with a total of 6 hours of audio coming from dysarthric speakers, and 9 hours of audio coming from control speakers.

TABLE I: Dysarthric speakers in the TORGO dataset

Severity	Speaker	# Utterances	% Prompt Overlap
Severe	F01	228	100%
	M01	739	99.1%
	M02	772	98.2%
	M04	659	98.2%
M/S	M05	610	98.9%
Moderate	F03	1097	95.7%
Mild	F04	675	98.6%
	M03	806	99.7%

The database consists of recordings of single words, sentences, and descriptions of contents in photographs. The speakers, their severity levels and the number of utterances has been summarized in Table I. The presence of only 957 unique texts among the total 16,394 text entries indeed indicates a significant level of overlap in the prompts used by the speakers. Table I also summarizes this as a percentage overlap between speakers. About 75% of the utterances are isolated word utterances.

Our work stays consistent with the protocol defined in Espana-Bonet et al. [15] which has been adopted in subsequent work [11], [12] which uses a leave-one-speaker-out evaluation protocol. Data from speaker F03 is used as validation data, and F04 in-case the training speaker is F03. In the leave-one-speaker-out protocol the training data consists of data from all speakers other than the target speaker. This includes data from the control speakers. The data from the target speaker is used as test data.

IV. Removing prompt overlap in TORGO

Our initial attempts involved reducing prompt overlap by manually selecting utterances. Instead of relying on heuristics, this section talks about a mixed integer linear programming (MILP) algorithm for data selection so that utterances with the same text prompt do not appear between the TORGO training and test speaker sets. The objective of Mixed Integer Linear Programming (MILP) is to maximize or minimize a linear objective function subject to bounds, linear and integer constraints on some variables.

Table II introduces some of the variables that are useful for the MILP optimization problem. As the table mentions, binary variables x_i and y_i are used to filter speaker specific TORGO train and test sets T_{train} and T_{test}. As a result of this filterting, the MILP algorithm produces S_{train} and S_{test}, which are the speaker specific output filtered data sets without prompt overlap. We call the resulting train and test dataset, S_{train} and S_{test}, the NP-TORGO (no-prompt overlap TORGO) dataset.

TABLE II: A summary of variables used for mixed-integer linear programming to produce the NP-TORGO dataset

Variable	Description
T_{train}	Original training set with prompt overlap (all but the target speaker).
T_{test}	Original test set with prompt overlap (only the target speaker).
f	Fraction of test data to retain.
S_{train}	Training set without prompt overlap, as a result of MILP selection.
S_{test}	Test set without prompt overlap, as a result of MILP selection.
x_i	A (binary) integer variable: 1 if prompt i is in S_{train}, 0 otherwise.
y_i	A (binary) integer variable: 1 if prompt i is in S_{test}, 0 otherwise.

The objective function for the MILP is to maximize the number of prompts in both training and test sets:

$$\text{Maximize} \quad \left(\sum_{i \in T_{\text{train}}} x_i + \sum_{i \in T_{\text{test}}} y_i \right), \tag{1}$$

subject to the following constraints,

1) No overlaps: If a prompt exists in the training set ($x_i = 1$), it cannot exist in the test set ($y_i = 0$) (and vice versa).
$$x_i + y_i \leq 1 \quad \forall i \in (T_{\text{train}} \cup T_{\text{test}}) \tag{2}$$
2) Each prompt in the train and test sets must either be 1 or 0:
$$x_i, y_i \in \{0, 1\} \tag{3}$$
3) A floor constraint is set to ensure a minimum number of prompts in the test set. This ensures that the size of S_{test} is at least a fraction f of the original test set T_{test}. At $f = 0$, no test prompts will be retained as the number of total prompts is maximized when no prompts are removed from the training set.
$$|S_{\text{test}}| = \sum_{i \in T_{\text{test}}} y_i \geq f \times |T_{\text{test}}| \tag{4}$$

To ensure a balanced distribution of both words and phrases in the train and test sets, the optimization problem was divided into two separate tasks: one for splitting isolated words and another for splitting phrases. The results from these two tasks were combined to construct the NP-TORGO dataset. The optimization problem is solved using the OR-Tools package developed by Google [20], particularly its `pywraplp` module. Additionally, the SCIP (Solving Constraint Integer Programs) algorithm, integrated within OR-Tools was employed to effectively solve the linear programming problem. SCIP is renowned for its capability in handling mixed-integer linear programming tasks.

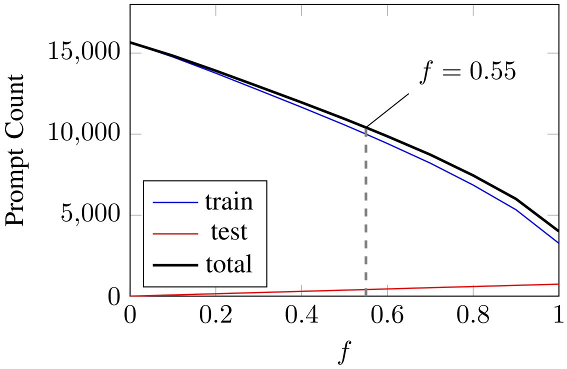

Fig. 3: Effect of f on the number of utterances retained in training and test set for target speaker M01.

In our early manual attempts to reduce overlap for each speaker we noticed a steep drop-off in training data quantity while trying to retain enough test data. Using heuristics across different speakers we found that a large part of the training data could be retained while preserving about 60% of the test data. Our work chooses $f = 0.55$ for the MILP algorithm so as to maintain at least 55% of the test speaker data. Figure 3 shows the variation in the prompt count with f for all prompts, training prompts and test prompts for one example speaker M01 in the TORGO dataset. The results of the train/test split optimization for the combined (containing both isolated words and sentences) dataset NP-TORGO are presented in Table III. It is notable that the optimization process achieved a ratio of retained prompts in the train set, ranging from 63.3% to 64.3%. Similarly, the test set maintained a percentage of retained prompts between 55.0% and 55.2%, meeting the constraint while maximizing the available data.

V. Experimental setup

This section talks about the acoustic models, language models and error-correction models used in the experimental study.

TABLE III: Controlling All Prompt Overlaps

Speaker	Train Set		Test Set	
	Before	After	Before	After
F01	16166	10232	228	126
F03	15319	9851	1075	592
F04	15727	10034	675	367
M01	15655	10002	739	407
M02	15628	9991	766	423
M03	15594	9976	800	442
M04	15742	10042	652	360
M05	15821	10077	573	316

A. *Acoustic models*

Our initial experiments investigated the impact of different variations of the wav2vec2 [21] architecture. Preliminary results showed strong baseline performance with cross-lingual representations. The 'wav2vec2-xlsr-53' [22] model consists of cross-lingual representations learned from 53 different languages using a semi-supervised objective function. This observation is similar to results reported by Hernandez et al [23].

For our acoustic model training our setup uses the HuggingFace training tools[3]. Per speaker acoustic models were trained with a connectionist temporal classification (CTC) [24] objective. The symbol set consists of a dictionary of 32 tokens encompassing a range of symbols including standard alphabetic characters ('a' through 'z') and a few special tokens. The acoustic models were trained using the Adam optimizer on an NVIDIA T4 GPU with a batch size of 4 and gradient accumulation every 2 steps. The learning rate was set to 10^{-4} with a linear warmup of 1000 steps. Regularization was applied using weight decay of 5×10^{-3}. While training was set to 20 epochs, the best model was chosen based on loss and word error rate scores on the validation set.

Additionally, to understand off-the-shelf naive model inference we use a 'wav2vec2-xlsr-53-en'[4] model trained on the Common-Voice dataset and we contrast this with the performance of naive inference with the Whisper-Large-V2 model.

B. *Language Models*

For training n-gram language models (LMs) our work uses the KenLM[25] toolkit. By default, KenLM utilizes modified Kneser-Ney smoothing including interpolation with weight backoff. All of our LM evaluations use different *tri-gram* language models. This includes an in-domain language model called **TORGO LM**, and out-of-domain language models called **NP-TORGO LM** and **Librispeech LM**.

For the **TORGO LM** and **NP-TORGO LM**, a different tri-gram LM is trained for each evaluation speaker. The training text consists of prompts from all other speakers in the training set except the test speaker from the TORGO and the NP-TORGO datasets respectively. For example, while choosing F01 as test speaker, F03 as default validation speaker, the training data consists of texts from all speakers except F01 and F03.

For the tri-gram **Librispeech LM**, text from the 360h training subset of the LibriSpeech corpus [13] is used. Librispeech is a read speech dataset based on LibriVox's audio books. There are fifty-eight thousand unique words, one-hundred thousand sentences, and 3.6 million tokens to train the language model.

Our analysis uses both the out-of-vocabulary (OOV) rate as well as perplexity. Table IV gives the OOV rate and perplexity for the test set for each speaker in the NP-TORGO dataset. In contrast Table V gives the perplexity and OOV rate for the original TORGO dataset. It is apparent that removing prompt overlap yields a high OOV rate for the NP-TORGO dataset.

TABLE IV: Out-of-domain LMs : Avg. Perplexity and OOV rate

Trained LM	Perplexity	OOV rate
LibriSpeech	3979.84	2.47%
NP-TORGO	462.97	59.99%

TABLE V: In-domain LM : Avg. Perplexity and OOV rate

Trained LM	Perplexity	OOV rate
TORGO	19.24	0.63%

C. *Cross-modal error-correction*

In order to study the impact of acoustic input for post-processing ASR, our study uses a recently proposed cross-modal error-correction (EC) model called Whispering-LLaMA [14][5]. As the name suggests, this EC model consists of the acoustic encoder from Whisper [26] with cross-modal attention to adapters in the LLaMA [27] model.

The authors introduce per layer adapter modules to the frozen LLaMA model. The first adapter variable A_i^L represents the adapter used in layer i to fine-tune the LLaMA model using a scaled dot product attention mechanism. The second adapter variable A_i^W refers to another adapter in layer i used to fuse Whisper features with the LLaMA model following an autoencoder mechanism. Each of these adapters have learnable matrices M_θ^i for the adapter variable A_i^L and M_{down}^i and M_{up}^i for the cross-modal fusion adapter variable A_i^W. The authors provide a parameter r which can be chosen as 8,16 or 32 to adjust the sizes of M_{down}^i and M_{up}^i. These respectively yield small, medium or large adapters. In our experiments we tried small(r=8) and medium (r=16) adapters.

The adapters are fine-tuned using instruction-fine tuning with n-best hypotheses generated by a Whisper model. Our experiments do not fine tune the Whisper model at all. In the original paper the authors use the Whisper-tiny model to generate weak hypotheses for the n-best list. For dysarthric speech it was found that we needed hypotheses generated by a larger Whisper-Large-V2 model. The number of hypotheses set to $n = 50$, instead of $n = 200$ used in the paper in order to save computation time. The acoustic features for cross-modal fusion were generated with the Whisper-Large-V2 model as well.

[3] www.huggingface.co

[4] https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english

[5] Code taken from https://github.com/Srijith-rkr/Whispering-LLaMA

The LLaMA model in this implementation has a maximum sequence length of 2048 tokens, and a maximum input length of 1000 tokens. We trained the adapters for 10 epochs, with a weight decay of 0.02, warmup set to 0, batch-size of 32 and a micro batch-size of 4 with a linear learning rate decay strategy.

VI. Experiments

This section summarizes the results of our experimental study. Table VI summarizes our baseline results with the wav2vec2-xlsr-53 model trained on dysarthric data with greedy CTC decoding. The results are presented on the TORGO dataset and the NP-TORGO dataset on a per speaker basis. For the TORGO dataset, results are presented on the full test set (100% test) and a reduced test set (55% test). The reduced test set (55% test) is identical to the test set obtained when constructing the NP-TORGO dataset.

TABLE VI: Baseline wav2vec2-xlsr53 with greedy decoding

Severity	**Speaker**	**TORGO**		**NP-TORGO**
		100% Test	**55% Test**	
Severe	F01	46.45%	47.27%	80.00%
	M01	40.72%	43.28%	85.07%
	M02	54.40%	55.21%	90.43%
	M04	64.50%	65.81%	93.75%
M/S	M05	51.04%	56.20%	90.38%
Moderate	F03	25.81%	25.43%	65.14%
Mild	F04	4.92%	4.46%	45.06%
	M03	3.17%	3.76%	45.21%

Tables VII and VIII break down the performance by severity on the TORGO and the NP-TORGO datasets respectively. Furthermore, the impact of various language models is presented at the isolated word (IW) level as well as the sentence (Sent.) level. Each row describes the impact on the word error rate (WER) of a particular language model. Table VIII additionally contains results for naive inference with off-the-shelf wav2vec2-xlsr-53 (labeled as w2v2-xlsr53-en) models as well as the Whisper-Large-V2 (labelled as Whisper-L-V2) model. Furthermore, the last two rows of Table VIII list the performance of the Whispering-LLaMA EC models with small (labelled as WL EC (s)) and medium adapters (labelled as WL EC (m)).

TABLE VII: Isolated word (IW) and Sentence (Sent.) performance using various language models on TORGO and NP-TORGO

TORGO Dataset						
Task	**Severe**		**Moderate**		**Mild**	
	IW	**Sent**	**IW**	**Sent**	**IW**	**Sent**
Baseline	54.9%	50.0%	51.4%	30.8%	7.0%	2.5%
TORGO LM	49.05%	37.9%	45.4%	23.4%	6.7%	1.6%
Librispeech LM	51.3%	42.9%	47.9%	24.5%	6.8%	1.8%

VII. Discussion

From the results in Table VI it appears that cross-lingual representations using the wav2vec2-xlsr-53 model provide a new state-of-the-art performance on TORGO (100% Test).

TABLE VIII: Performance on NP-TORGO

NP-TORGO Dataset						
Task	**Severe**		**Moderate**		**Mild**	
	IW	**Sent**	**IW**	**Sent**	**IW**	**Sent**
w2v2-xlsr53-en	114.8%	73.9%	95.9%	40.2%	41.7%	8.42%
Whisper-L-V2	108.1%	56.3%	88.4%	21.8%	28.9%	5.16%
Baseline	93.9%	83.8%	93.9%	68.6%	68.4%	35.6%
NP-TORGO LM	96.2%	80.0%	95.2%	61.7%	73.6%	33.8%
Librispeech LM	93.9%	78.5%	90.0%	58.8%	63.5%	27.8%
WL EC (s)	92.9%	79.1%	86.1%	28.8%	55.1%	13.8%
WL EC (m)	93.0%	62.9%	84.6%	29.1%	50.0%	11.3%

The WER performance does not appear to be drastically affected when presented on the reduced test set (55% test). The improvements are especially apparent when looking at the mild speakers, with their WER performance in the 3-5% range.

Prompt overlap leads to a significant over-estimation of the performance, where the model easily starts to memorize all of the prompts. This is consistent with the observations made by Yue et al. [12]. On removing prompt overlap, the same acoustic models perform rather poorly overall on the NP-TORGO dataset. The impact is evident when it comes to isolated word recognition. For speaker M05 for example, the words recognized are acoustically close to the ground truth, but are not the exact word. With the drastic drop in performance for NP-TORGO, there is strong evidence to suggest that prompt overlap leads to an over-estimation of the WER performance on TORGO.

A further breakdown of the results on the TORGO and NP-TORGO datasets appears in Tables VII and VIII. In Table VII in-domain tri-gram LMs with backoff are seen to have a significant impact on isolated word WER performance for speakers with severe and moderate dysarthria. The impact of out of domain language models (such as Librispeech LM) appears limited. On the other hand in NP-TORGO performance in Table VIII suggests that out-of-domain language models have a more significant role to play when there is no prompt-overlap between train and test utterances. Out-of-domain language models are seen to have a limited impact on the isolated word performance, and understandably a larger impact on the sentence level performance for severe, moderate and mild speakers. In addition the prompts taken from the NP-TORGO training set to train the NP-TORGO LM seem to hurt the performance on the isolated word recognition task.

Naive inference results in Table VIII for off-the-shelf models such as Whisper-Large-V2 and the Wav2vec2-xlsr-53-en model show poor performance on the isolated word task, but provide some strong results for moderate and naive speakers on the sentence level task. Performance on mildly dysarthric speakers seems to be outstanding, but less so for those with severe dysarthria. Results with our baseline model where dysarthric speaker data without prompt overlap is used to train our ('wav2vec2-xlsr-53') acoustic model seem to yield moderate performance improvements in the case of speakers with severe dysarthria on the isolated word task. Error-correction using the Whispering-LLaMa models show some promise. They are able to improve on the isolated word task performance for severe speakers, but degrade sentence level performance compared to naive inference. The performance for mild speakers seems to get worse for sentence level correction compared to naive inference.

VIII. Conclusion

Our work shows that state-of-the-art models even after post-processing with powerful error-correction models are still not ready for servicing dysarthric speakers in telehealth and in-person healthcare settings. To evaluate existing models this work presented a principled method for constructing a dysarthric ASR dataset as a subset of TORGO without prompt-overlap using linear programming. Results were presented to understand the impact of performing this split. Furthermore, our results show that cross-modal error-correction models based on LLMs hold promise for dysarthric ASR, but they struggle when it comes to isolated word scenarios. In order to improve isolated and sentence level word performance on this challenging baseline, future work will look at incorporating phonetic information into error-correction models along with careful data augmentation.

Acknowledgment

The authors would like to acknowledge support from the Khoury West Coast Research Fund and the Khoury SEED Grant from the Khoury College of Computer Sciences from Northeastern University that financially supported Macarious Hui and Jinda Zhang for this work.

References

[1] C. Handberg and A. K. Voss, "Implementing augmentative and alternative communication in critical care settings: Perspectives of healthcare professionals," *Journal of Clinical Nursing*, vol. 27, no. 1-2, pp. 102–114, 2018.

[2] A. Mohan, M. Chakraborti, K. Eng, N. Kushaeva, M. Prpa, J. Lewis, T. Zhang, V. Geisler, and C. Geisler, "A powerful and modern AAC composition tool for impaired speakers," *Interspeech 2024: Show and Tell Demo*, 2024.

[3] D. Mulfari, L. Carnevale, A. Galletta, and M. Villari, "Edge computing solutions supporting voice recognition services for speakers with dysarthria," in *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*. IEEE, 2023, pp. 231–236.

[4] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[5] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[6] Y. Li, P. Chen, P. Bell, and C. Lai, "Crossmodal ASR error correction with discrete speech units," *arXiv preprint arXiv:2405.16677*, 2024.

[7] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The torgo database of acoustic and articulatory speech from speakers with dysarthria," *Language Resources and Evaluation*, vol. 46, pp. 523–541, 2012.

[8] X. Menendez-Pidal, J. B. Polikoff, S. M. Peters, J. E. Leonzio, and H. T. Bunnell, "The nemours database of dysarthric speech," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 3. IEEE, 1996, pp. 1962–1965.

[9] M. Hasegawa-Johnson, "Universal access automatic speech recognition project," 2006.

[10] M. Nicolao, H. Christensen, S. Cunningham, P. Green, and T. Hain, "A framework for collecting realistic recordings of dysarthric speech-the homeservice corpus," in *Proceedings of LREC 2016*. European Language Resources Association, 2016.

[11] E. Hermann and M. M. Doss, "Dysarthric speech recognition with lattice-free mmi," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6109–6113.

[12] Z. Yue, F. Xiong, H. Christensen, and J. Barker, "Exploring appropriate acoustic and language modelling choices for continuous dysarthric speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6094–6098.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[14] S. Radhakrishnan, C.-H. Yang, S. Khan, R. Kumar, N. Kiani, D. Gomez-Cabrero, and J. Tegnér, "Whispering LLaMA: A cross-modal generative error correction framework for speech recognition," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10 007–10 016. [Online]. Available: https://aclanthology.org/2023.emnlp-main.618

[15] C. Espana-Bonet and J. A. Fonollosa, "Automatic speech recognition with deep neural networks for impaired speech," in *Advances in Speech and Language Technologies for Iberian Languages: Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016, Proceedings 3*. Springer, 2016, pp. 97–107.

[16] N. M. Joy and S. Umesh, "Improving acoustic models in torgo dysarthric speech database," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 3, pp. 637–645, 2018.

[17] S. Dutta, S. Jain, A. Maheshwari, S. Pal, G. Ramakrishnan, and P. Jyothi, "Error correction in asr using sequence-to-sequence models," *arXiv preprint arXiv:2202.01157*, 2022.

[18] Y. Leng, X. Tan, W. Liu, K. Song, R. Wang, X.-Y. Li, T. Qin, E. Lin, and T.-Y. Liu, "Softcorrect: Error correction with soft detection for automatic speech recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 034–13 042.

[19] C. Park, Y. Jang, S. Lee, J. Seo, K. Yang, and H.-S. Lim, "Pictalky: Augmentative and alternative communication for language developmental disabilities," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, 2022, pp. 17–27.

[20] L. Perron and V. Furnon, "OR-Tools," Google, 2023. [Online]. Available: https://developers.google.com/optimization/

[21] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[22] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[23] A. Hernandez, P. A. Pérez-Toro, E. Nöth, J. R. Orozco-Arroyave, A. Maier, and S. H. Yang, "Cross-lingual self-supervised speech representations for improved dysarthric speech recognition," *arXiv preprint arXiv:2204.01670*, 2022.

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[25] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*, 2011, pp. 187–197.

[26] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[27] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.